

FIG. 1

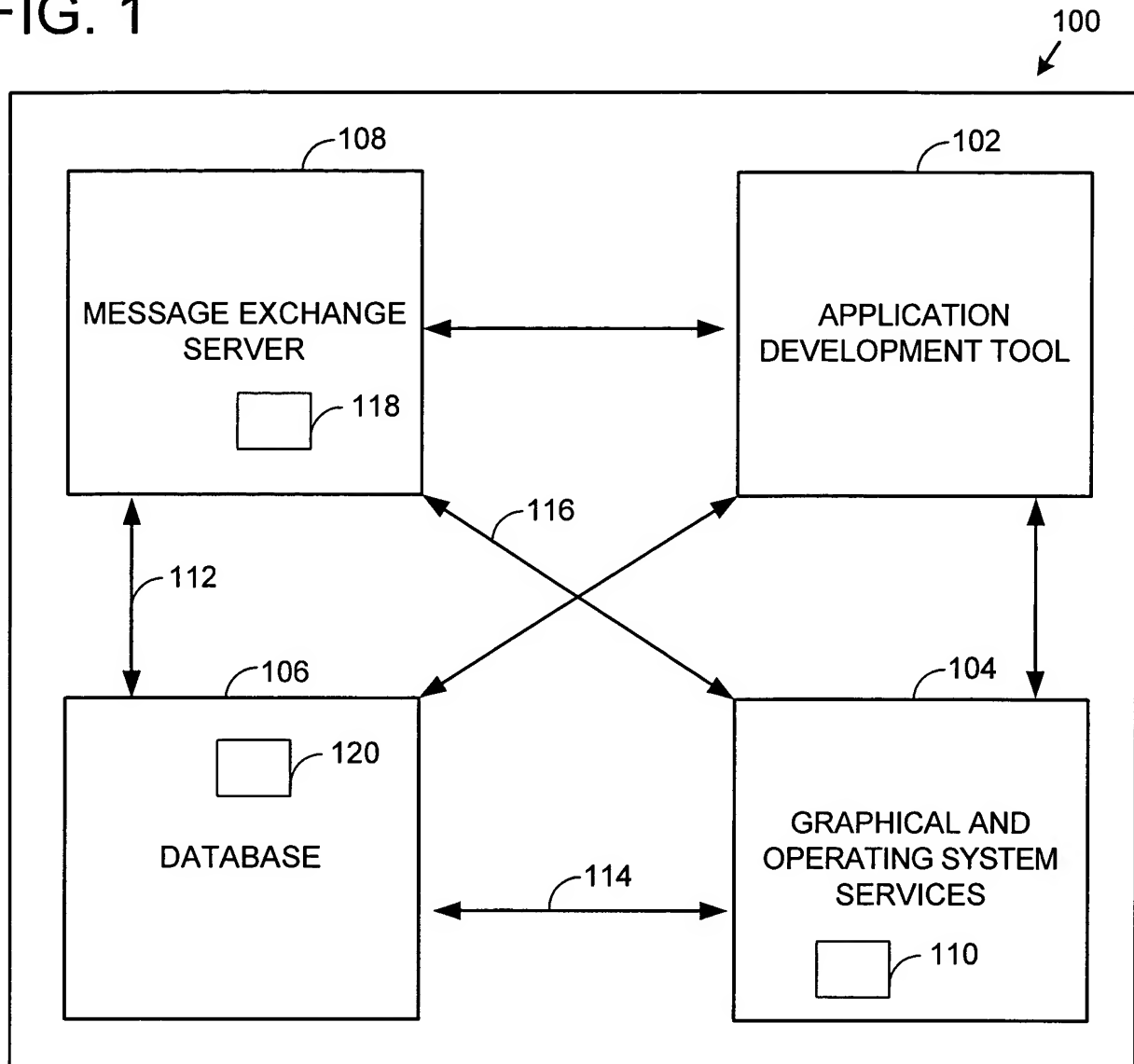


FIG. 2

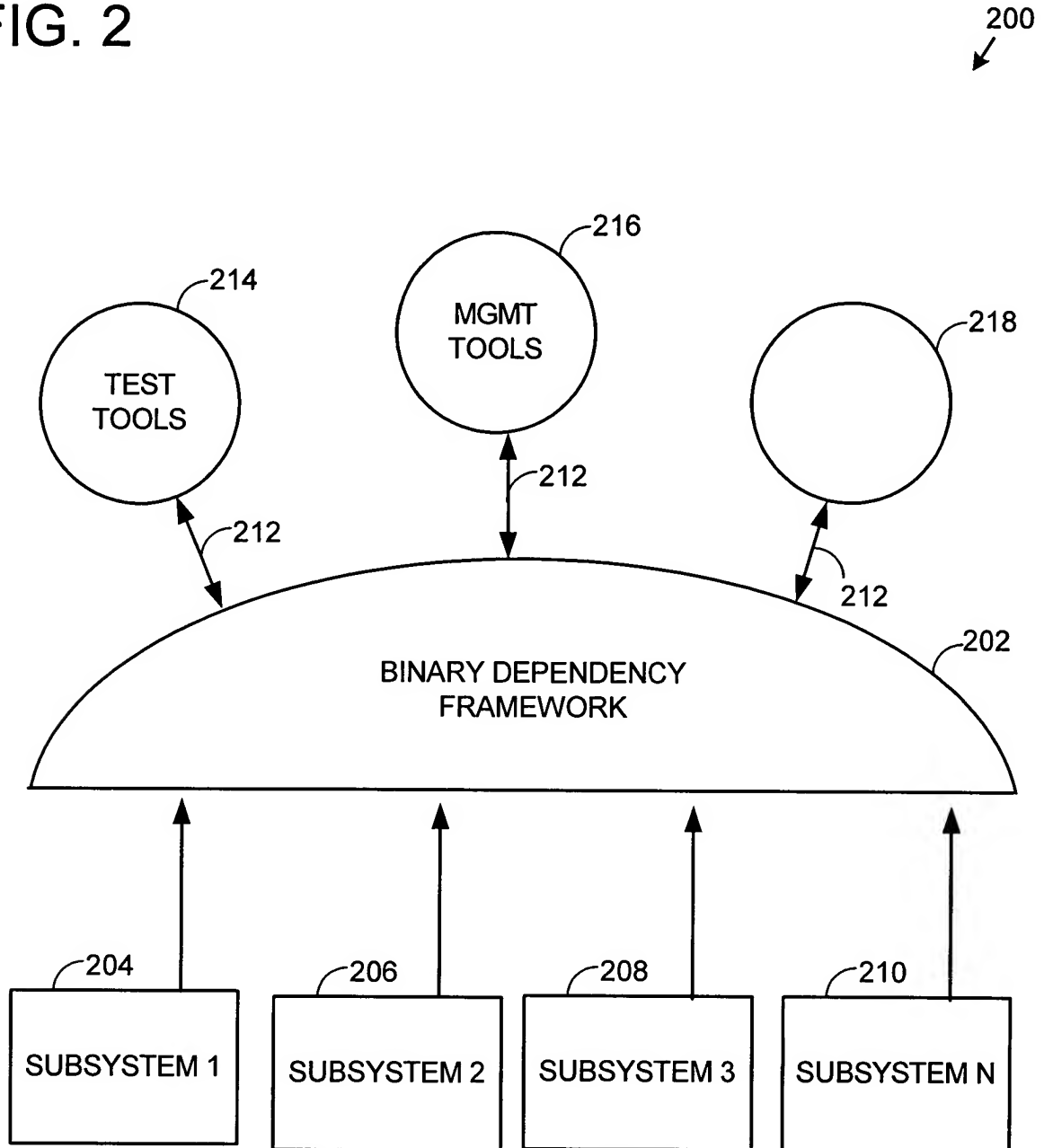


FIG. 3

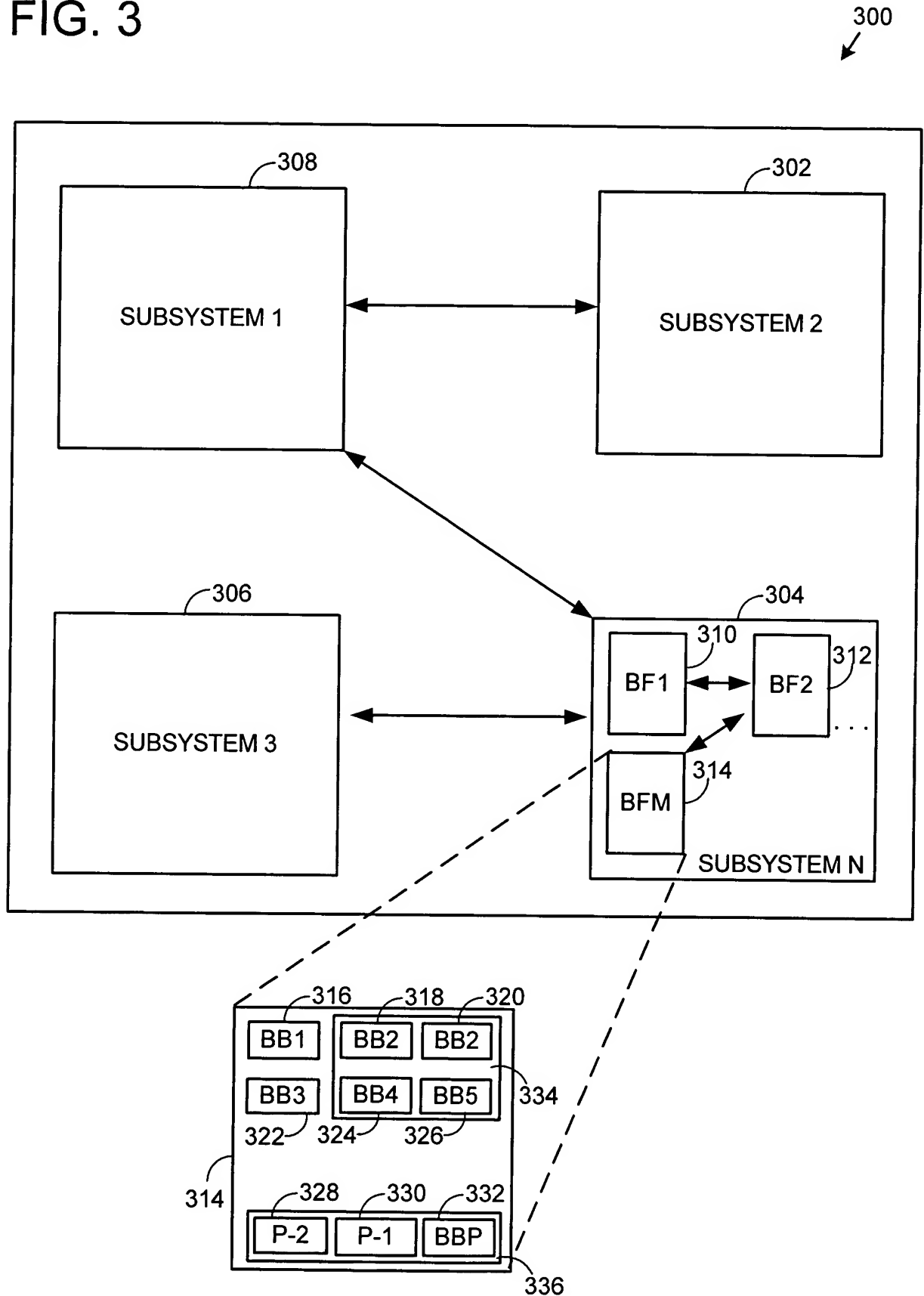


FIG. 4

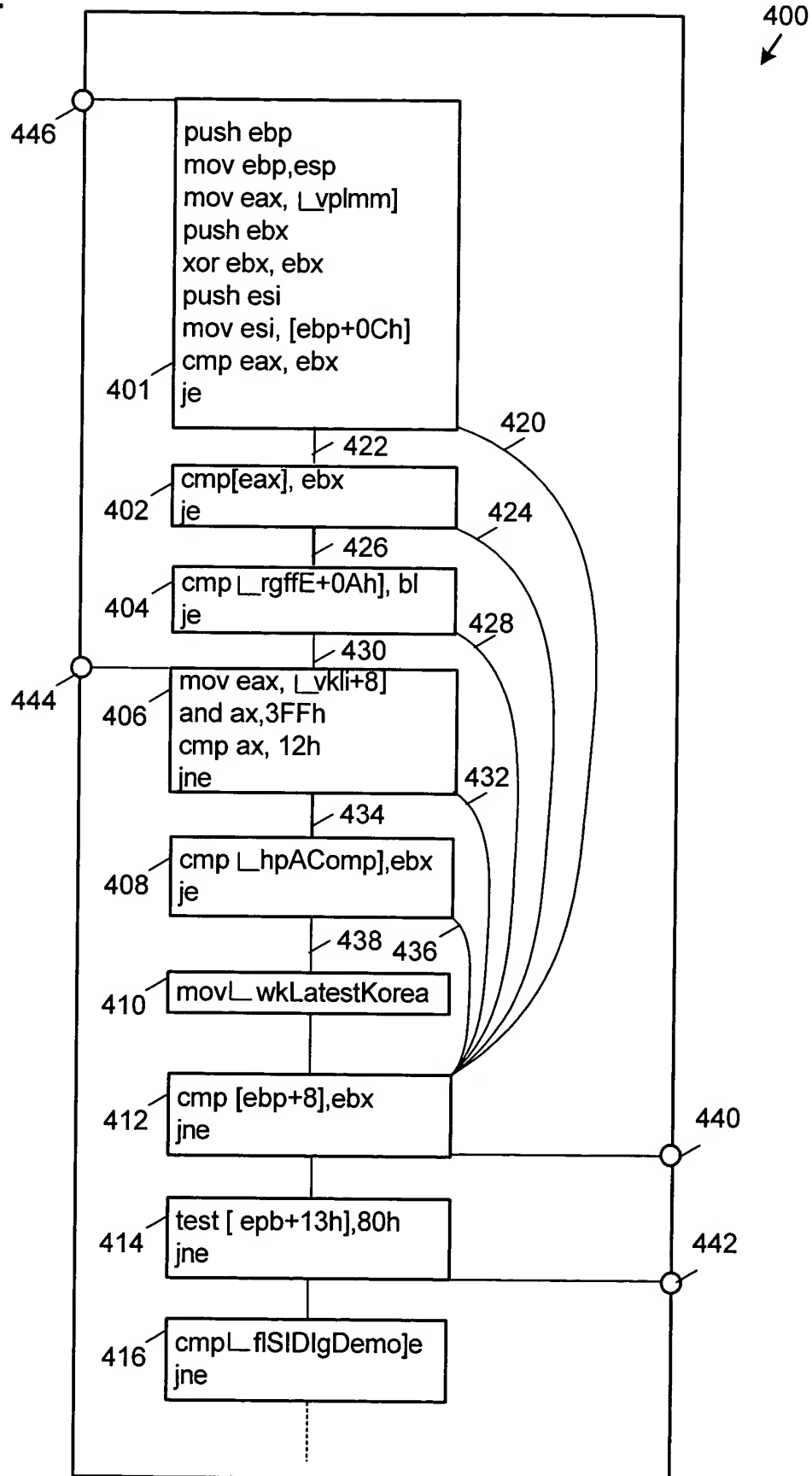


FIG. 5

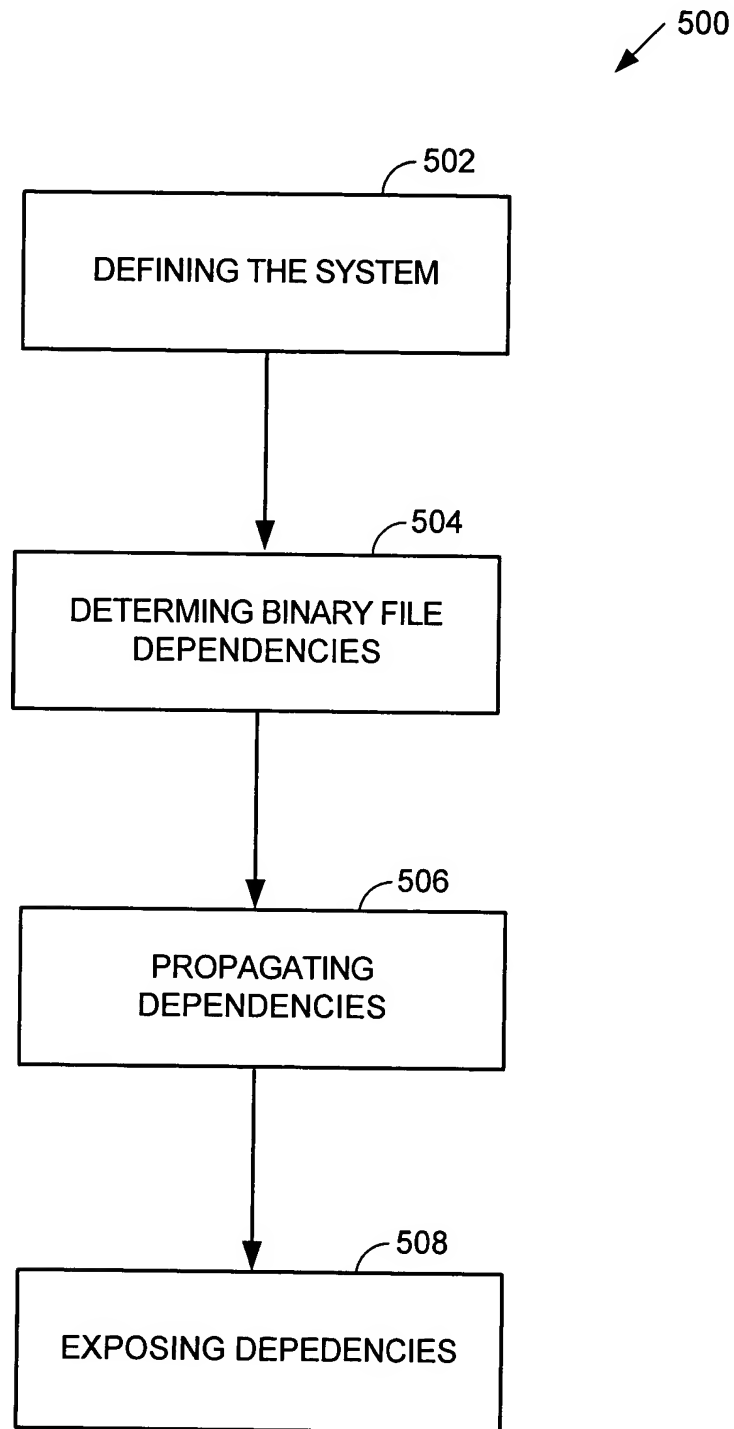


FIG. 6

600

The diagram shows an XML structure for a system named "magsys". The root element is <system name="magsys">, which is pointed to by reference numeral 602. Inside this system, there are several subsystems, each with its own set of binaries. The first subsystem is "magellan", pointed to by 604, which contains binaries "coverage.dll", "covercmd.exe", "magcore.dll", and "magtraces.dll". The second is "vulcan", the third is "vc", and the fourth is "windows". Each subsystem is closed with </subsystem>. The entire structure is closed with </system>. Reference numerals 606, 608, 610, 612, and 614 point to specific elements: 606 points to the first <subsystem> tag, 608 points to the first <binary> tag, 610 points to the first file attribute, 612 points to the last <binary> tag, and 614 points to the last file attribute.

```
<system name = "magsys">
  <subsystem name = "magellan" file = "mag.xml">
    <binary name = "coverage.dll" file = "coverage.xml"/>
    <binary name = "covercmd.exe" file = "covercmd.xml"/>
    <binary name = "magcore.dll" file = "magcore.xml"/>
    <binary name = "magtraces.dll" file = "magtraces.xml"/>
  </subsystem>
  <subsystem name = "vulcan" file = "vulcan.xml">
    <binary name = "vulcan23.dll" file = "vulcan23.xml" />
    <binary name = "vuldyn.exe" file = "vuldyn.xml" />
    <binary name = "vuldynpxy.dll" file = "vuldynpxy.xml" />
    <binary name = "vulutil.dll" file = "vulutil.xml" />
  </subsystem>
  <subsystem name = "vc" file = "vc.xml">
    <binary name = "mspdb71.dll" file = "mspdb71.xml" />
    <binary name = "msvcr71.dll" file = "msvcr71.xml" />
    <binary name = "msvcpx71.dll" file = "msvcpx71.xml" />
    <binary name = "msobj71.dll" file = "msobj71.xml" />
  </subsystem>
  <subsystem name = "windows" file = "windows.xml">
    <binary name = "kernel32.dll" file = "kernel32.xml" />
    <binary name = "nt.dll" file = "nt.xml" />
    <binary name = "user32.dll" file = "user32.xml" />
    <binary name = "gdi32.dll" file = "gdi32.xml" />
  </subsystem>
</system>
```

FIG. 7

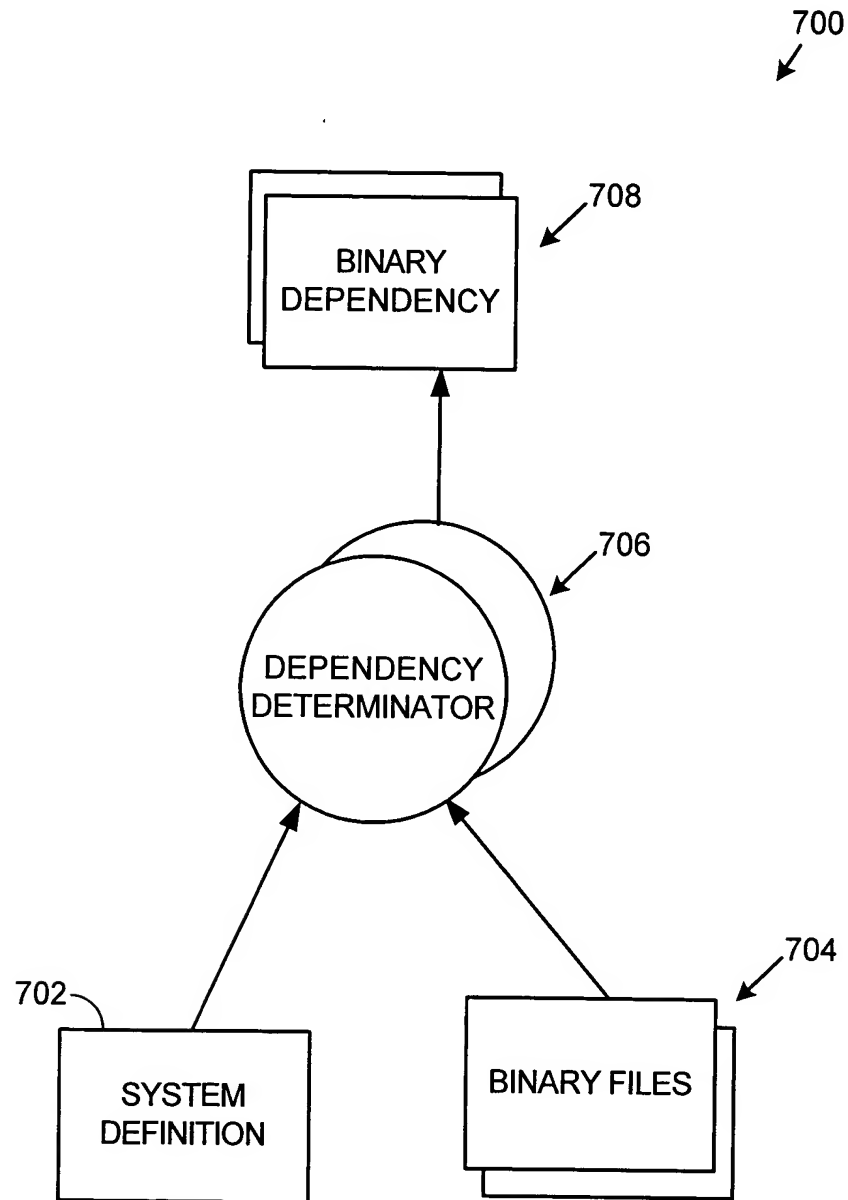


FIG. 8

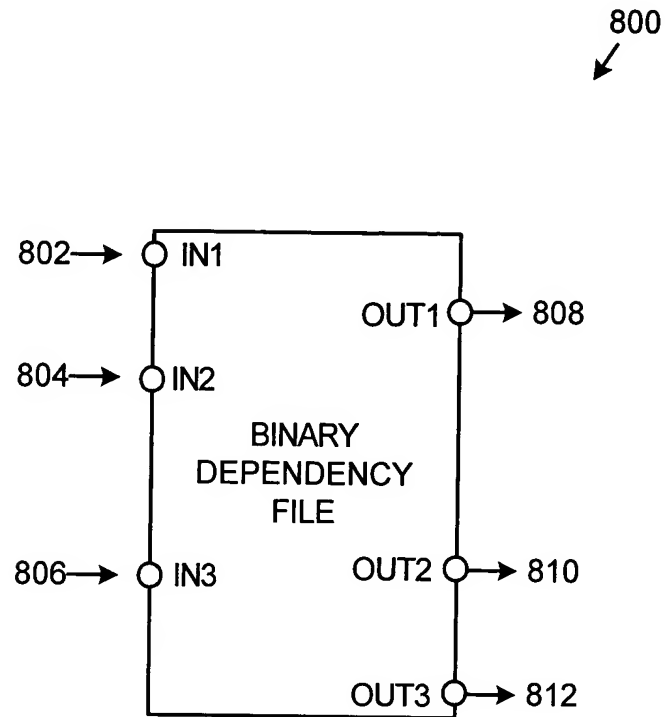


FIG. 9

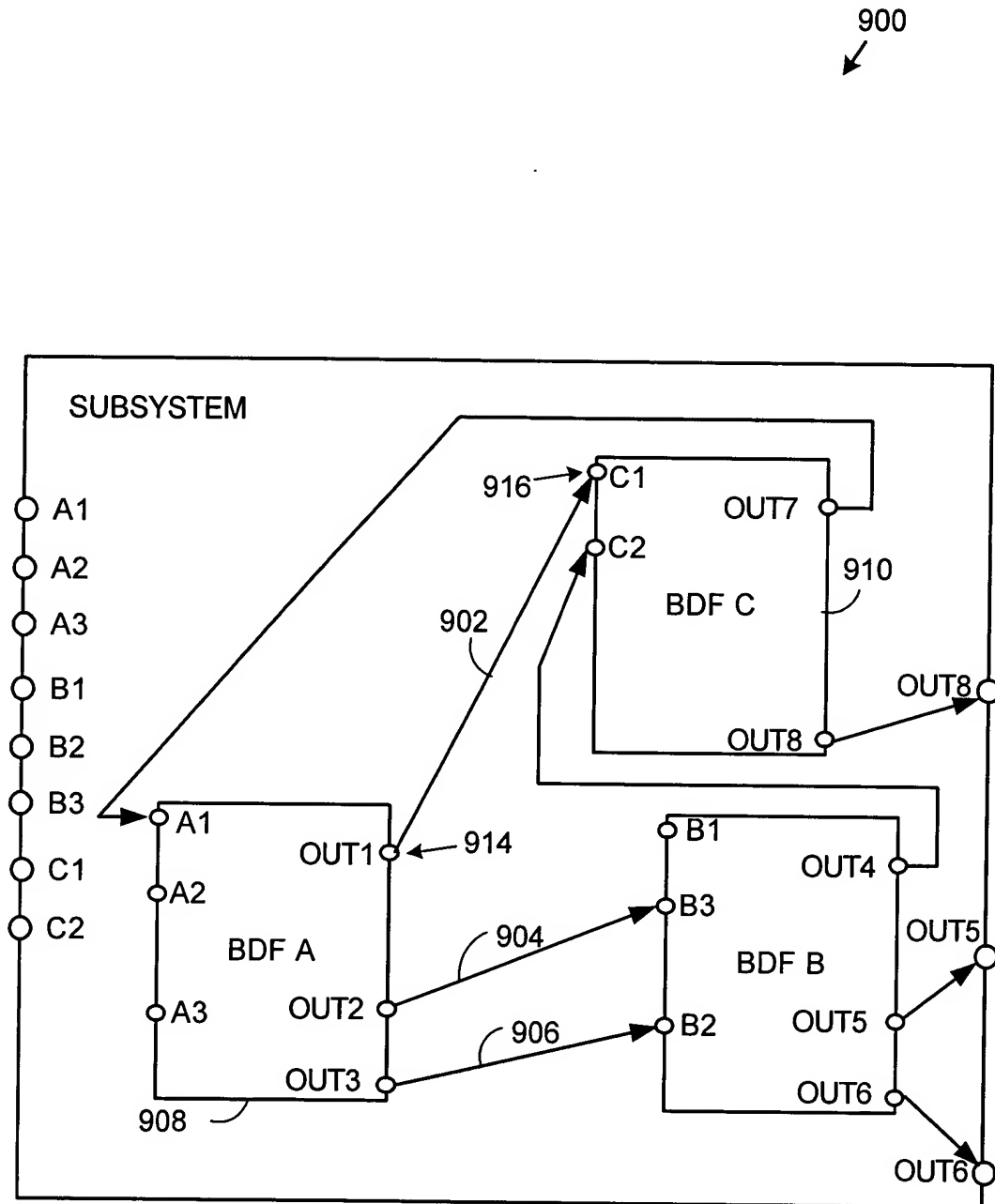


FIG. 10

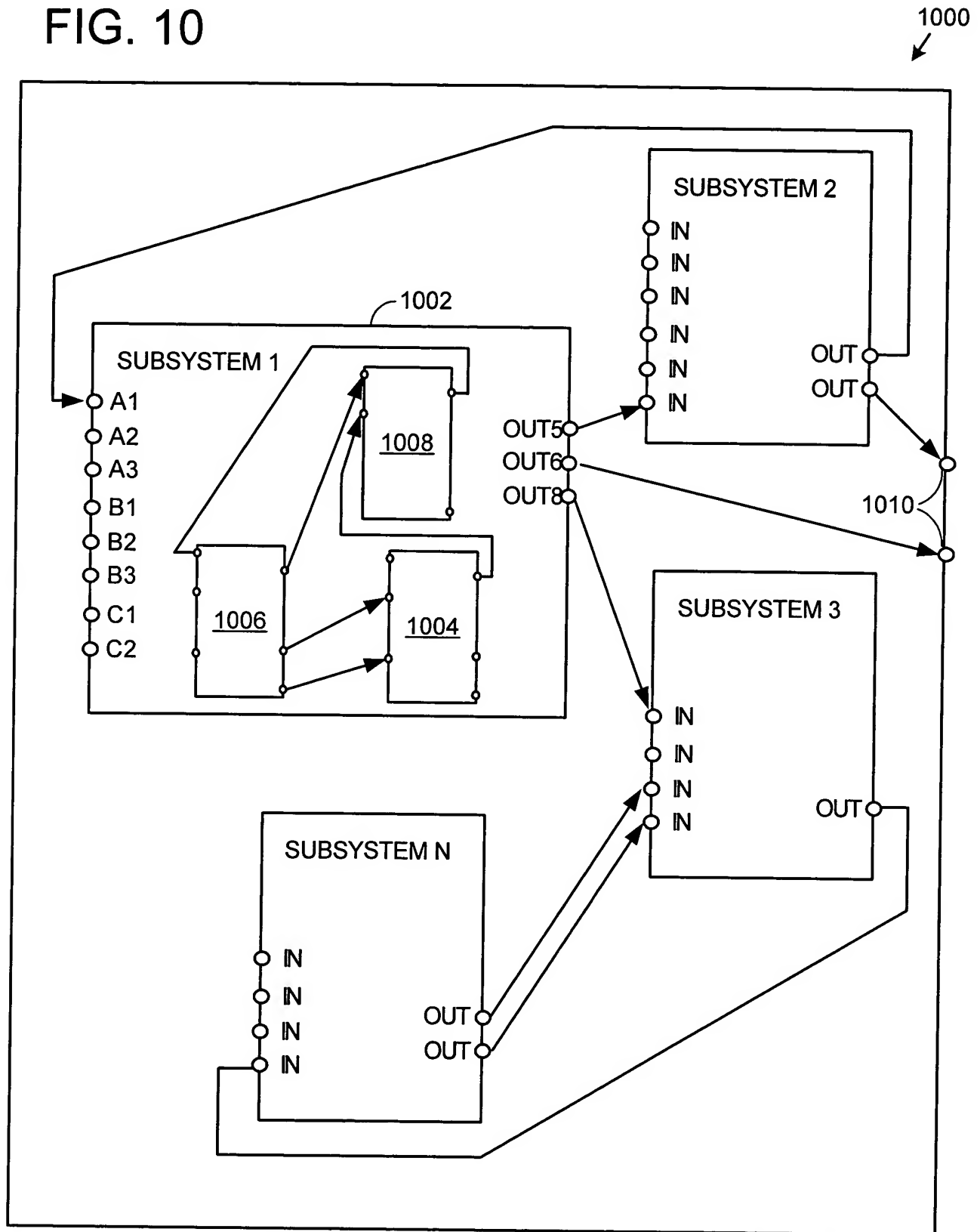


FIG. 11

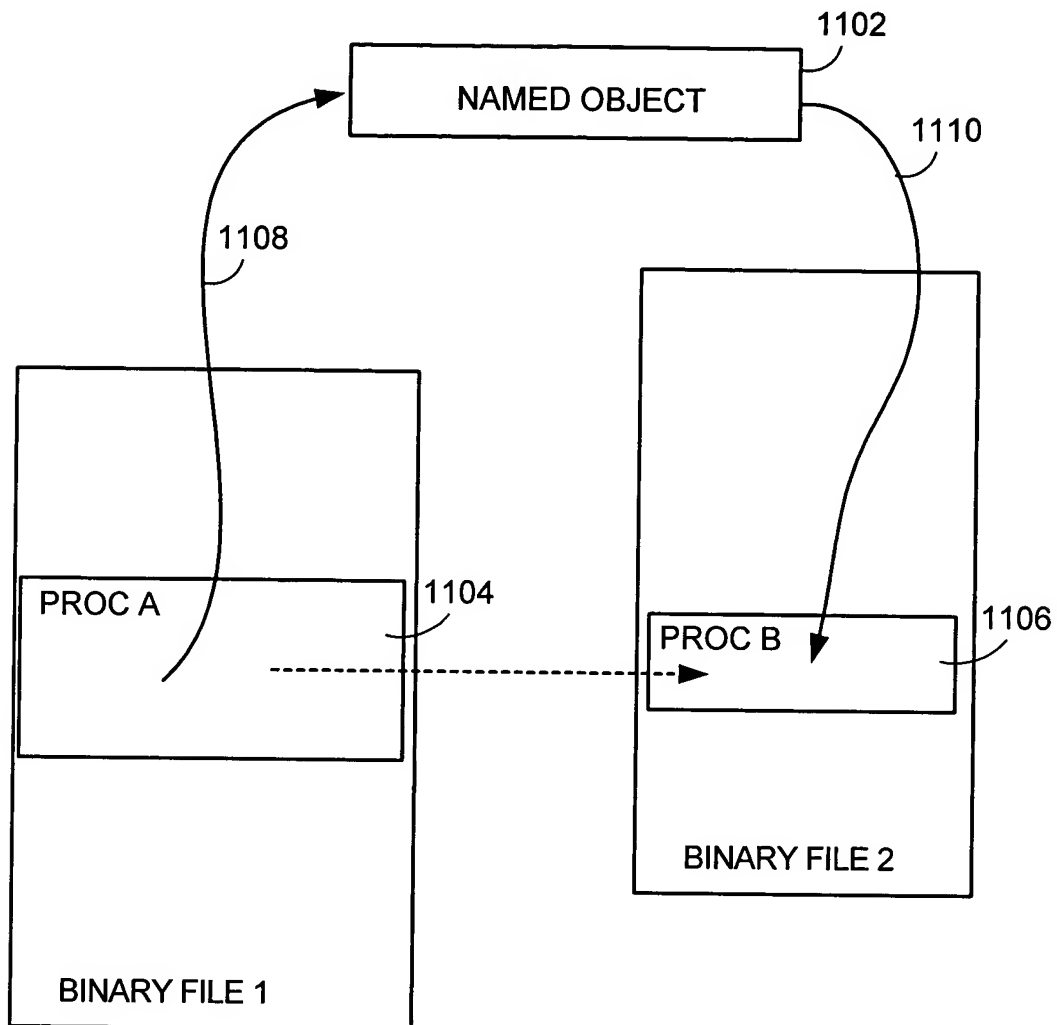


FIG. 12

1200
↙

```
CLASS SYSTEM ← 1202
    METHOD CREATEFROM ( *SYSTEMDEFFILE, *GUIDMAPFILE )
    METHOD DESTROY ( ) ← 1208
    METHOD NAME ( ) ← 1210
    METHOD FILE ( ) ← 1212
    METHOD GUIDMAPPINGFILE ( ) ← 1214
    METHOD FIRSTSUBSYSTEM ( ) ← 1216
    METHOD NEXTSUBSYSTEM ( ) ← 1218
    METHOD FIRSTNAMEDOBJECT ( ) ← 1220
    METHOD NEXTNAMEDOBJECT ( ) ← 1222
    METHOD FINDNODE ( *BINARYNAME, *FNNAME ) ← 1224
    METHOD FINDBINARY ( *BINARYNAME ) ← 1226
    METHOD FINDNAMEDOBJECT ( *NAMEDOBJECT ) ← 1228
CLASS SUBSYSTEM ← 1230
    METHOD NAME ( ) ← 1232
    METHOD TYPE ( )
    METHOD GETSYSTEM ( ) ← 1236
    METHOD FIRSTBINARY ( ) ← 1238
    METHOD NEXTBINARY ( ) ← 1240
    METHOD GETASSEMBLY ( )
CLASS BINARY ← 1244
    METHOD NAME ( ) ← 1246
    METHOD XMLFILE ( ) ← 1248
    METHOD DIRECTORY ( ) ← 1250
    METHOD GETSUBSYSTEM ( ) ← 1252
    METHOD FIRSTINPUT ( ) ← 1254
    METHOD NEXTINPUT ( ) ← 1256
    METHOD CREATEILBINARY ( )
CLASS NODE ← 1260
    METHOD NAME ( ) ← 1262
    METHOD GETFIRSTCALLER ( ) ← 1264
    METHOD GETNEXTCALLER ( ) ← 1268
    METHOD GETFIRSTCALLEE ( ) ← 1270
    METHOD GETNEXTCALLEE ( ) ← 1272
CLASS ILBINARY ← 1274
CLASS ASSEMBLY ← 1276
CLASS NAMEDOBJECT ← 1278
CLASS FILTER ← 1280
CLASS PROCEDURE ← 1282
CLASS PARAMETER ← 1284
```

FIG. 13

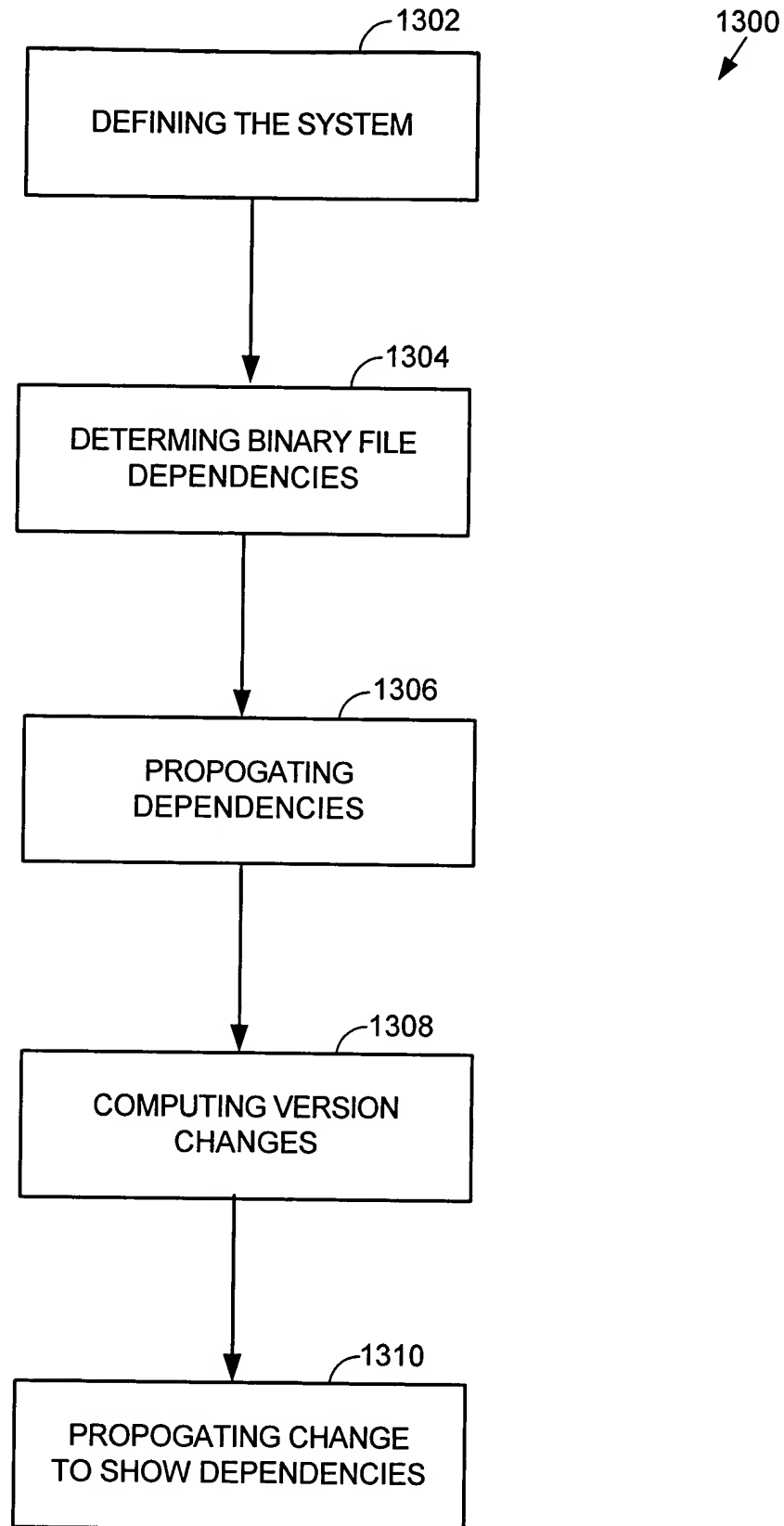


FIG. 14

1404

Input: System Definition File: System, Subsystem Sets
 Binary Information File: Entry-Exit dependencies
 Output: A set of affected entry points for Binary and Subsystem, and System

Algorithm:

```

    for each subsystem s in System ← 1408
    {
      for each binary b in s ← 1402
      {
        mark blocks changed (modified or added) ← 1404
        mark entry points of b that can reach a
        changed block as affected ← 1406
      }
    }

    while no new entry point is marked affected ← 1410
    {
      for each binary b in Subsystem ← 1412
      {
        for each exit point x in b not marked affected and
        connected to an entry point marked affected ← 1414
        {
          mark all entry points of b dependent on x as affected ← 1416
        }
      }
    }

    while no new entry point is marked affected
    {
      for each subsystem s in System ← 1418
      {
        for each exit point x in s not marked affected and
        connected to an entry point marked affected ← 1420
        {
          mark all entry points of s dependent on x as affected ← 1422
        }
      }
    }
  
```

FIG. 15

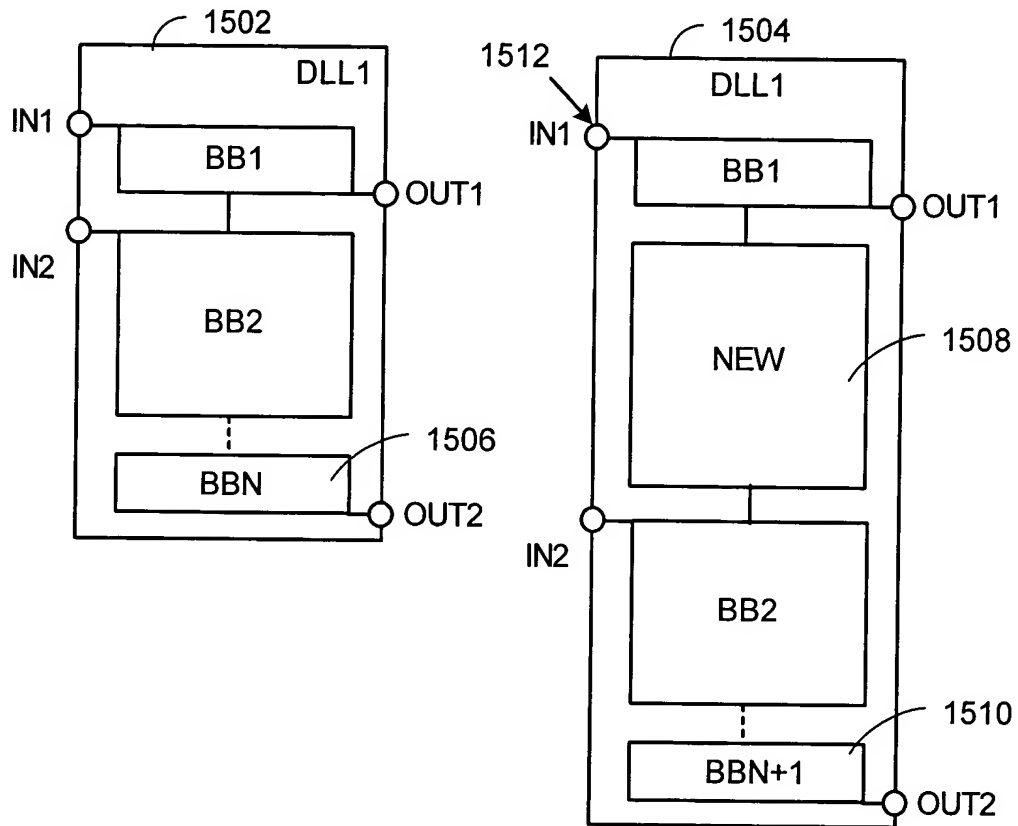


FIG. 16

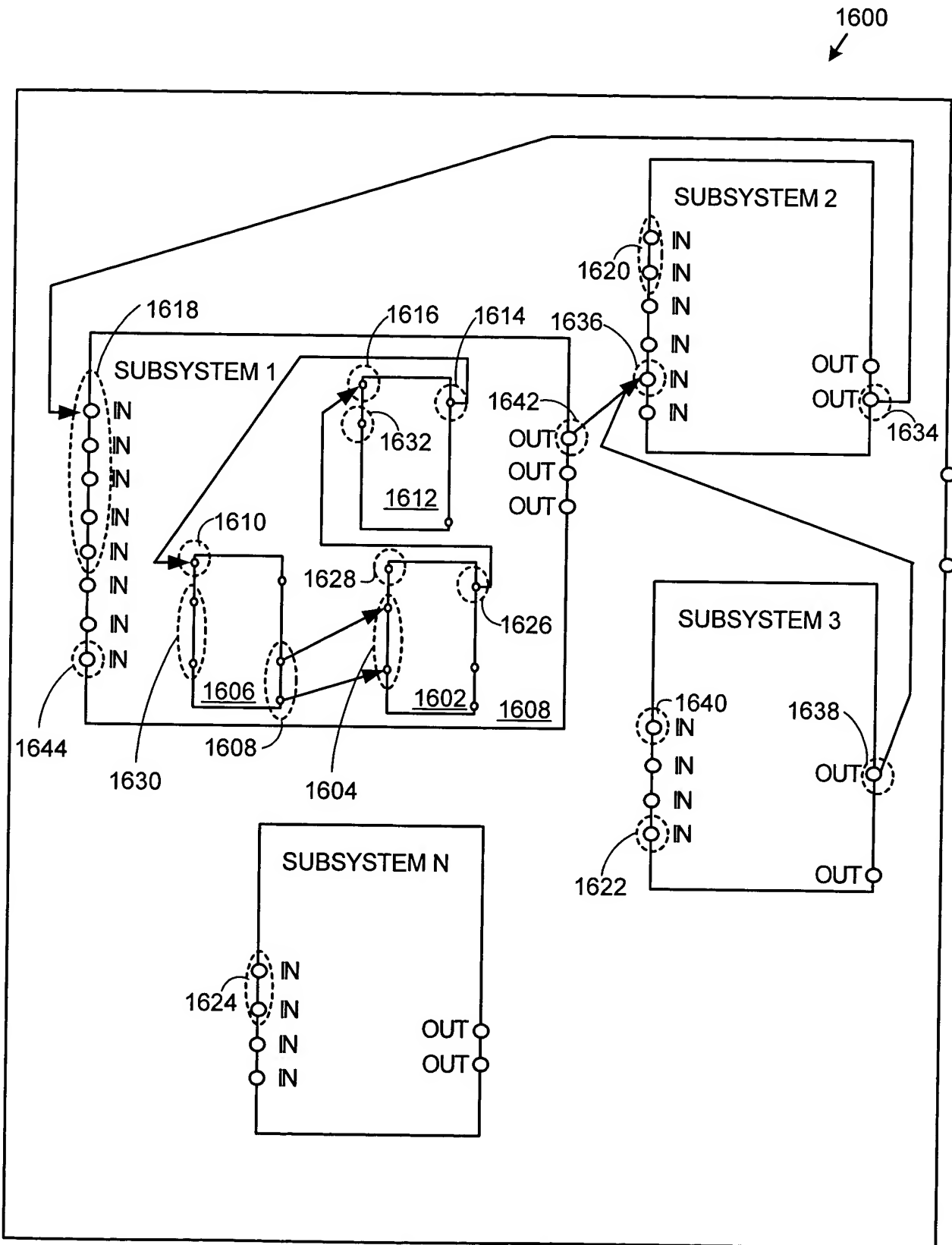
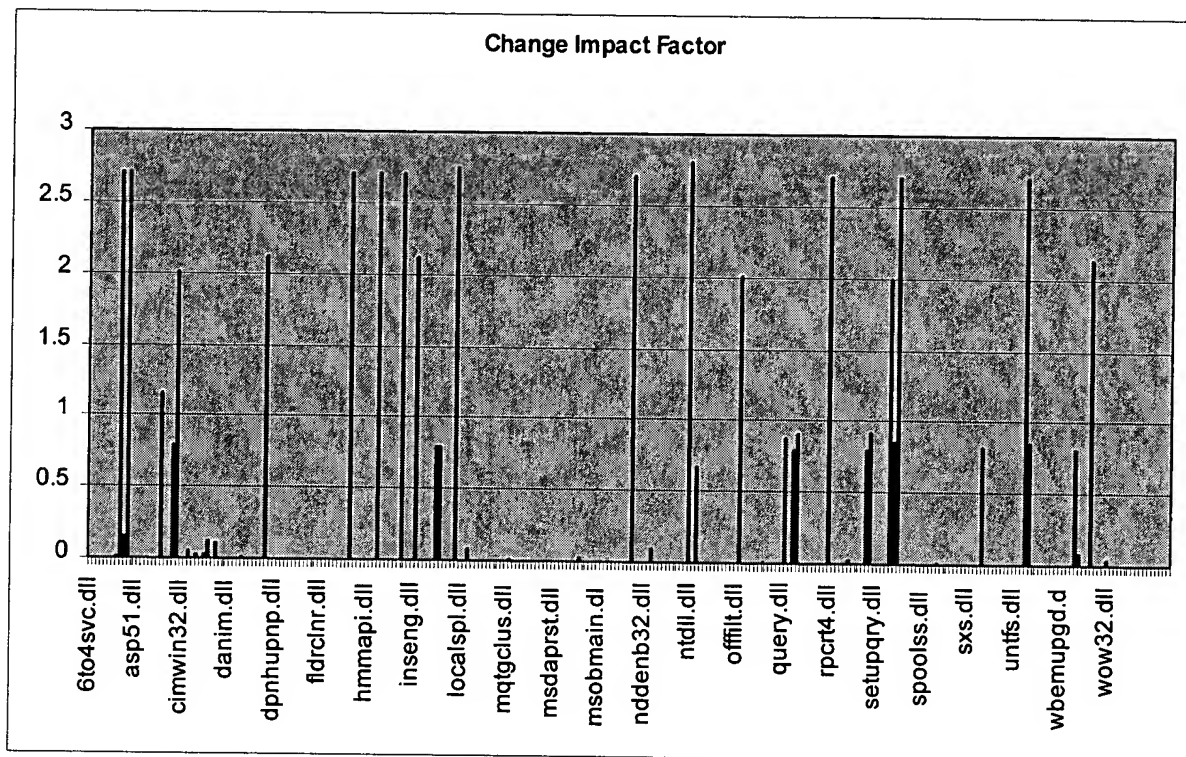


FIG. 17



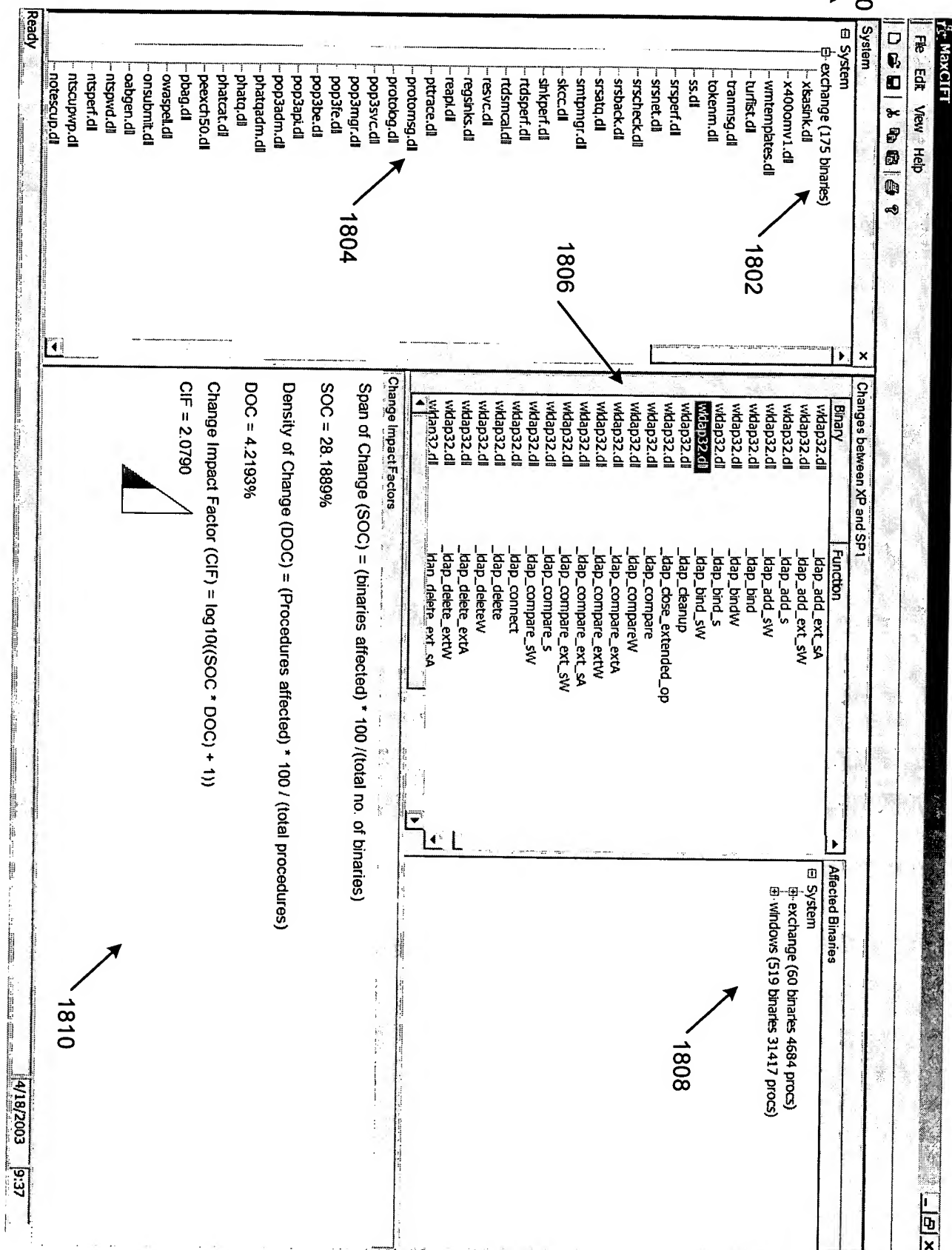


FIG. 19

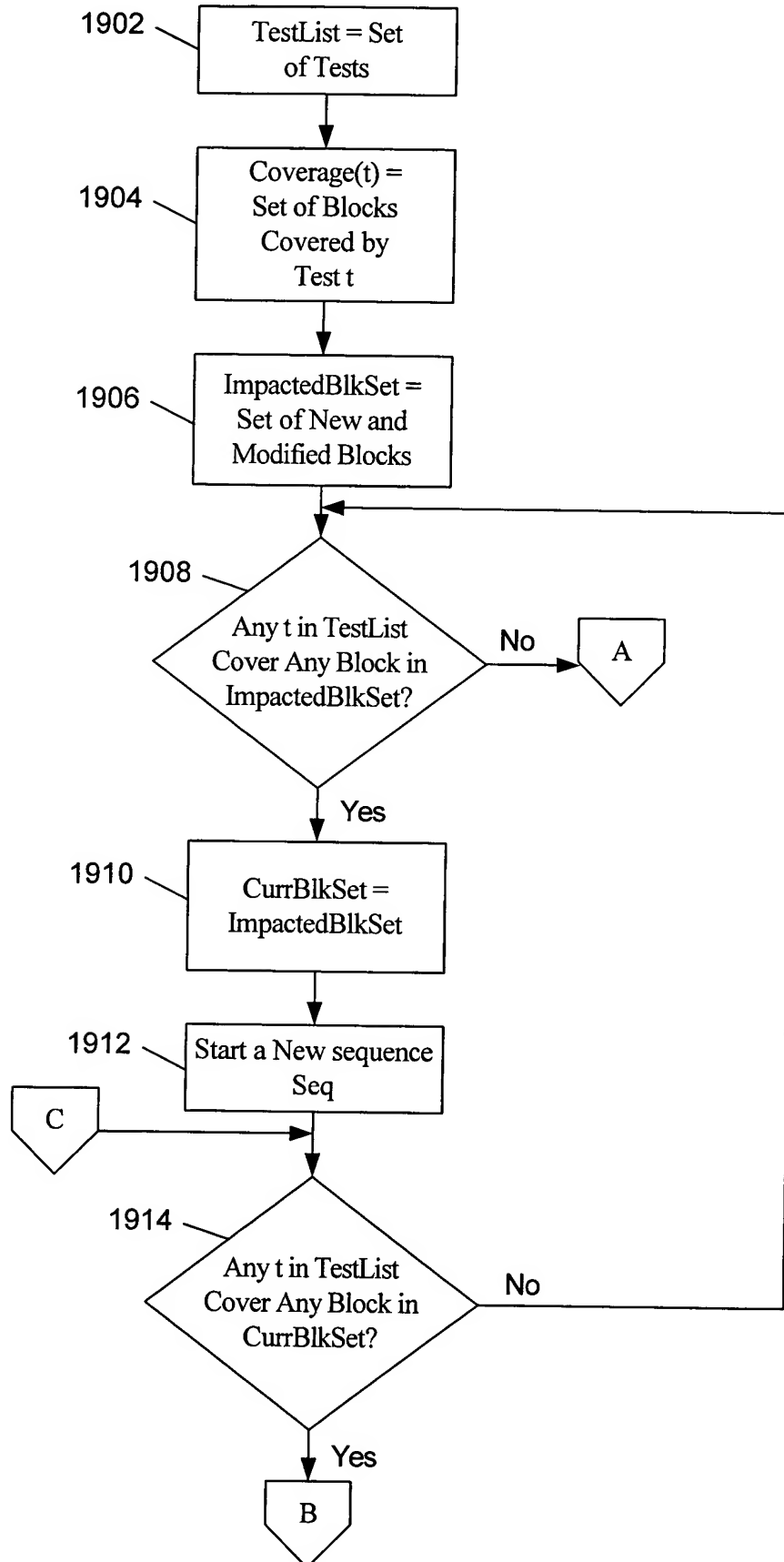


FIG. 20

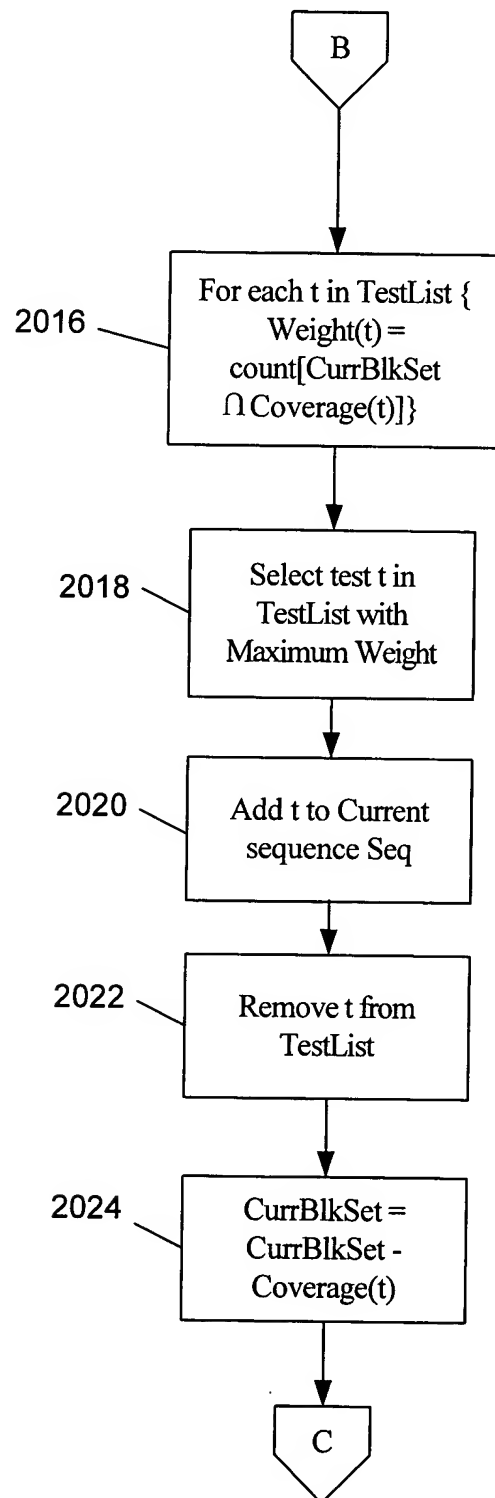
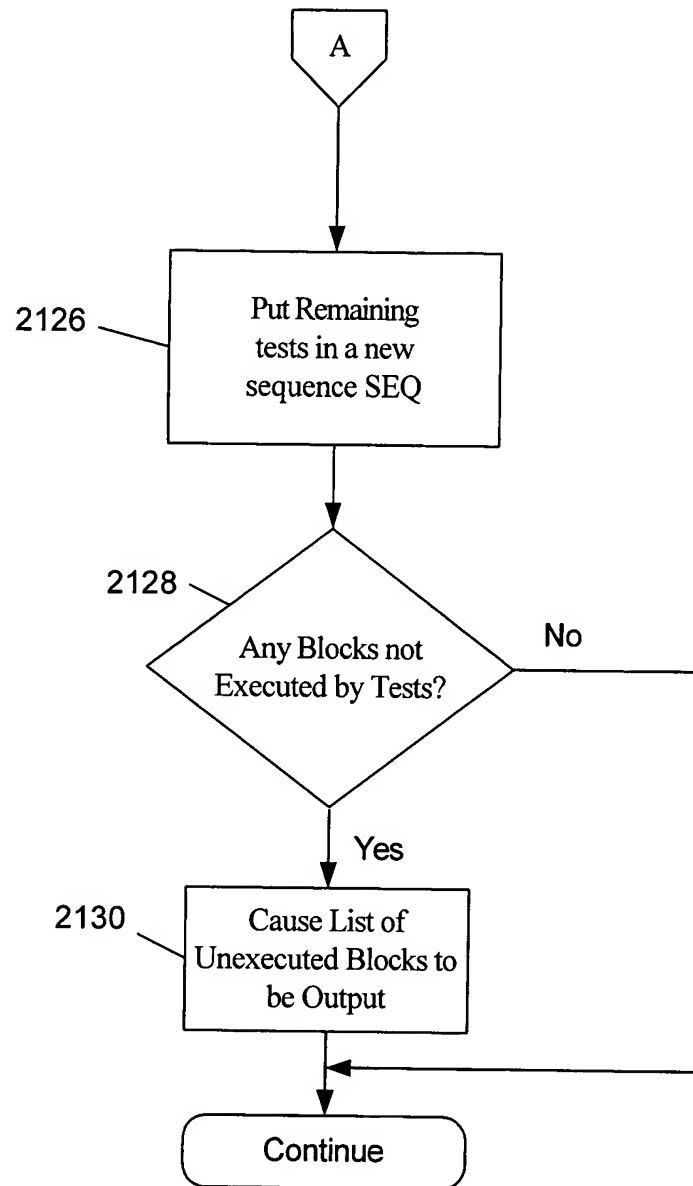


FIG. 21



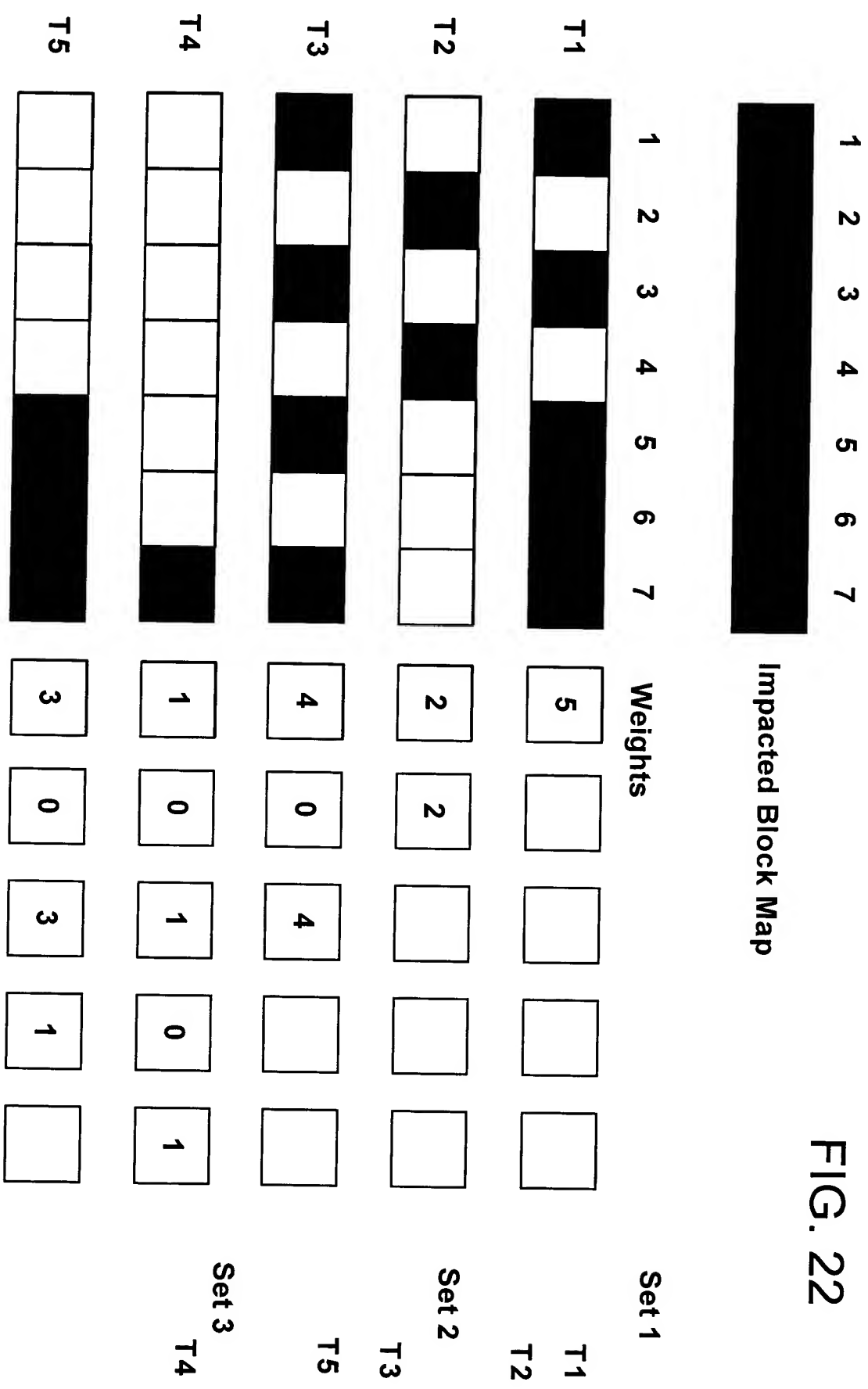


FIG. 23

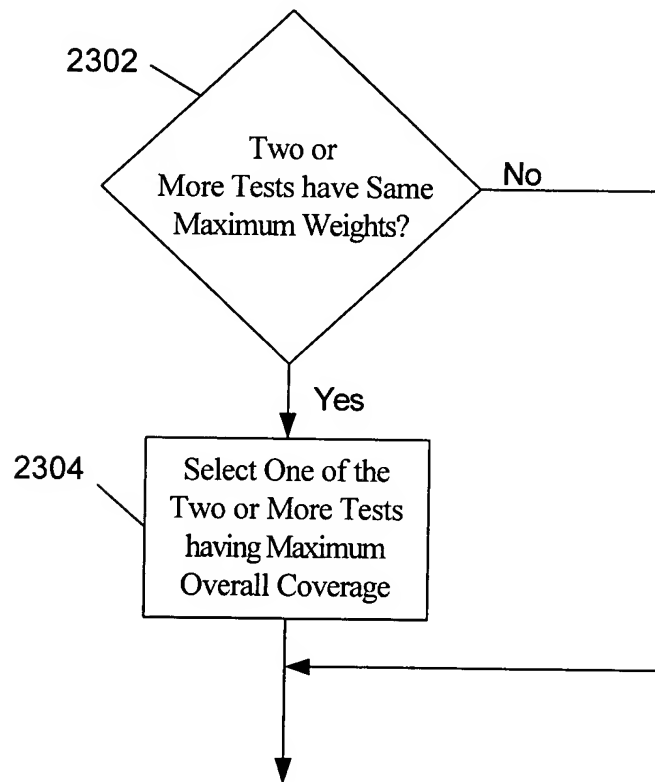


FIG. 24

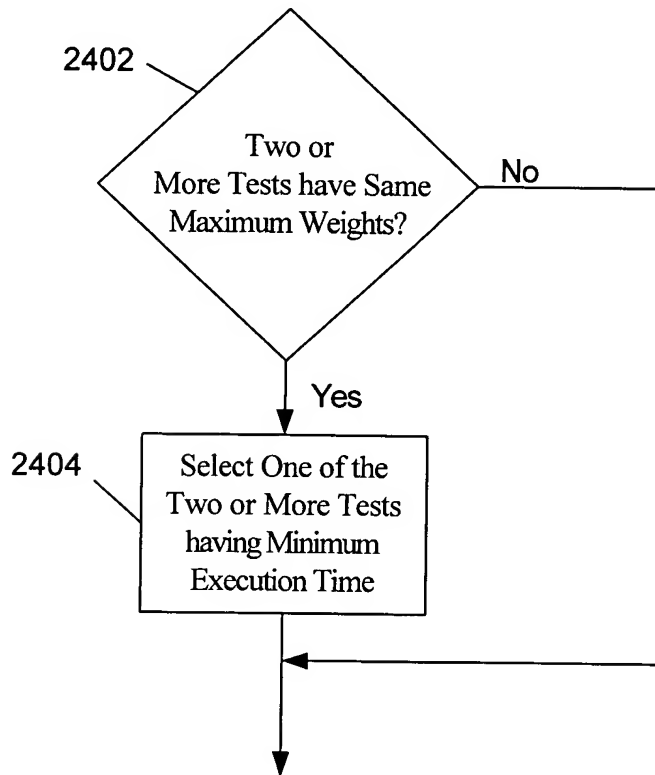


FIG. 25

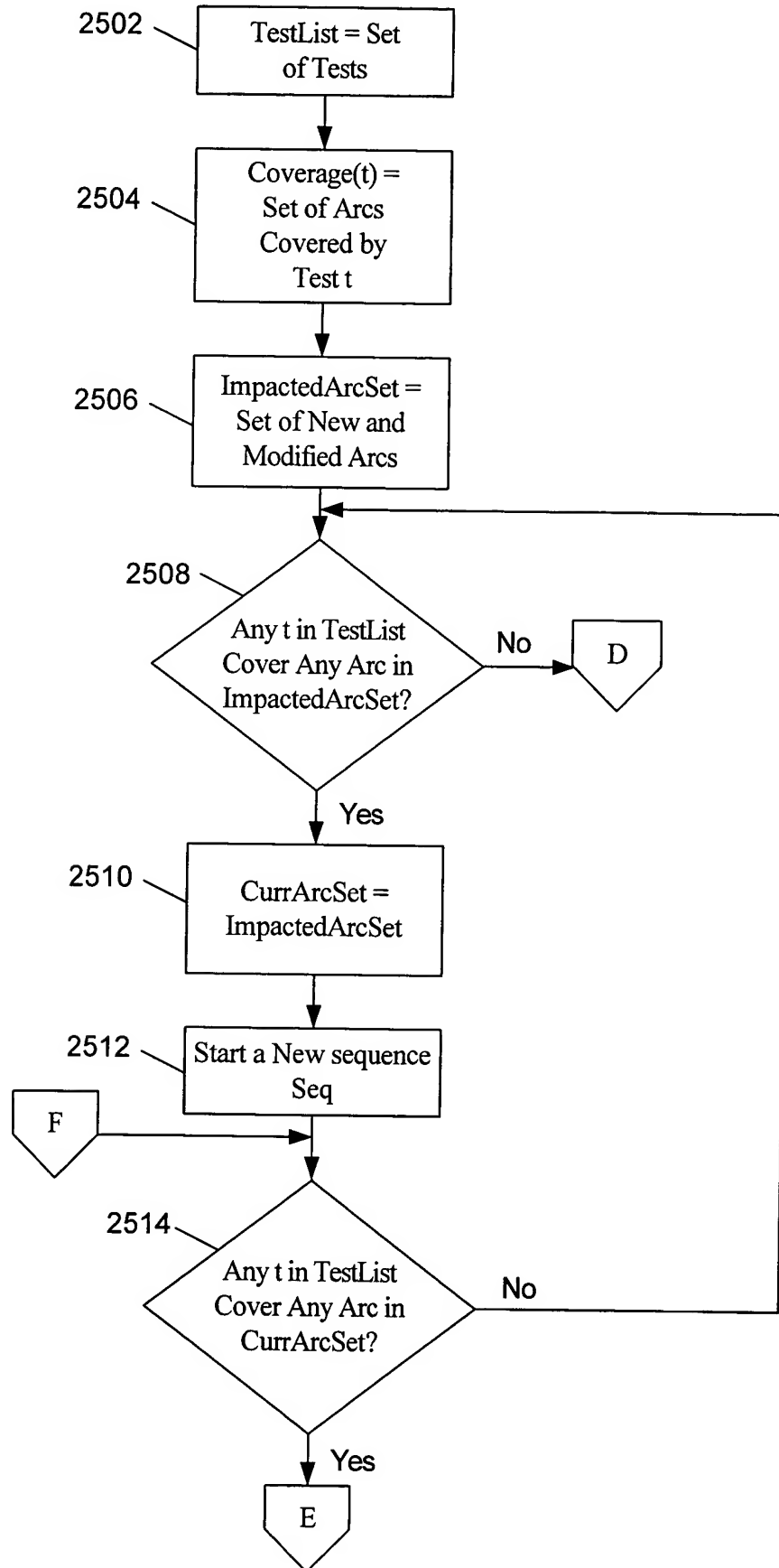


FIG. 26

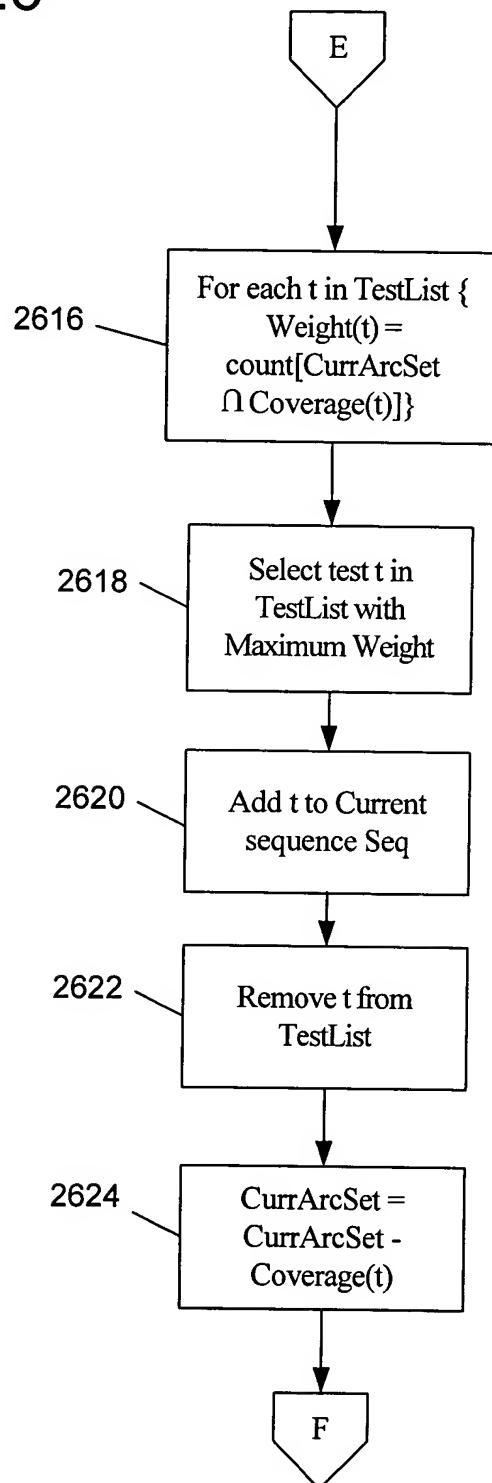


FIG. 27

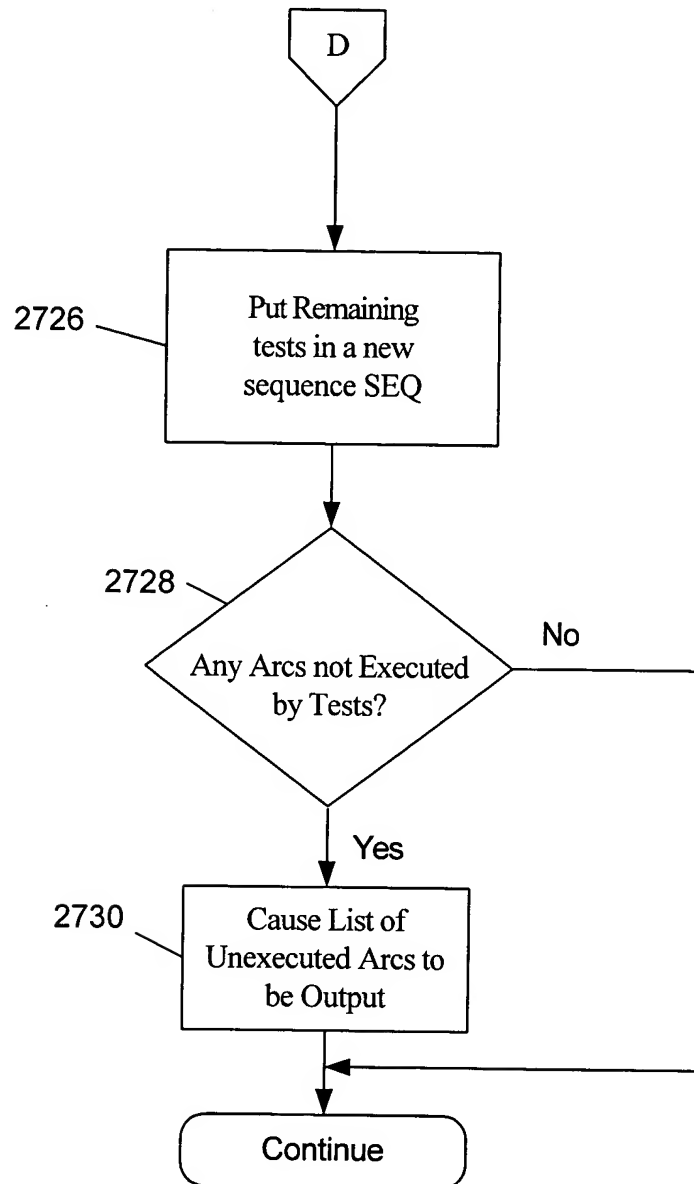
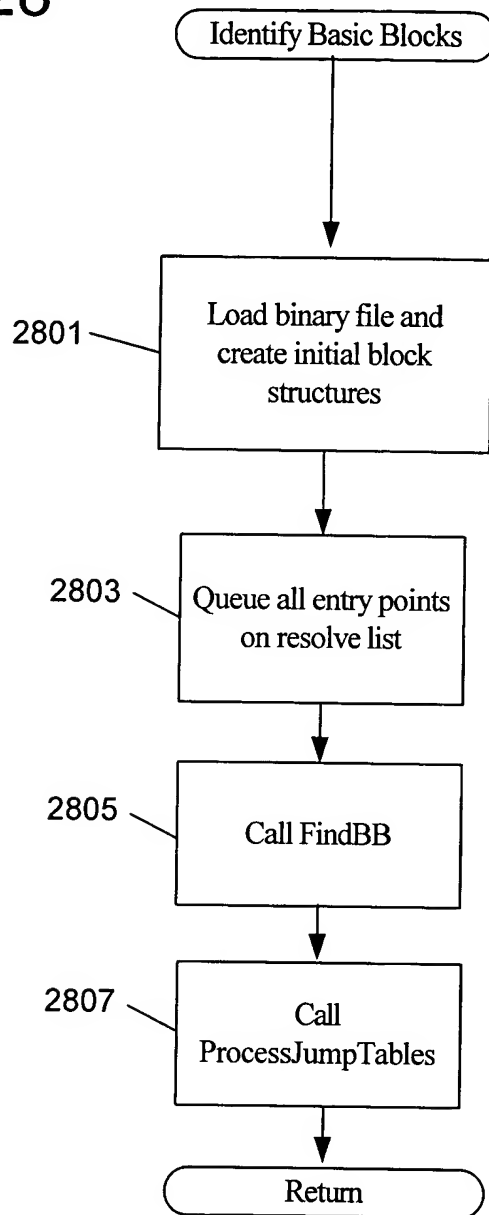


FIG. 28



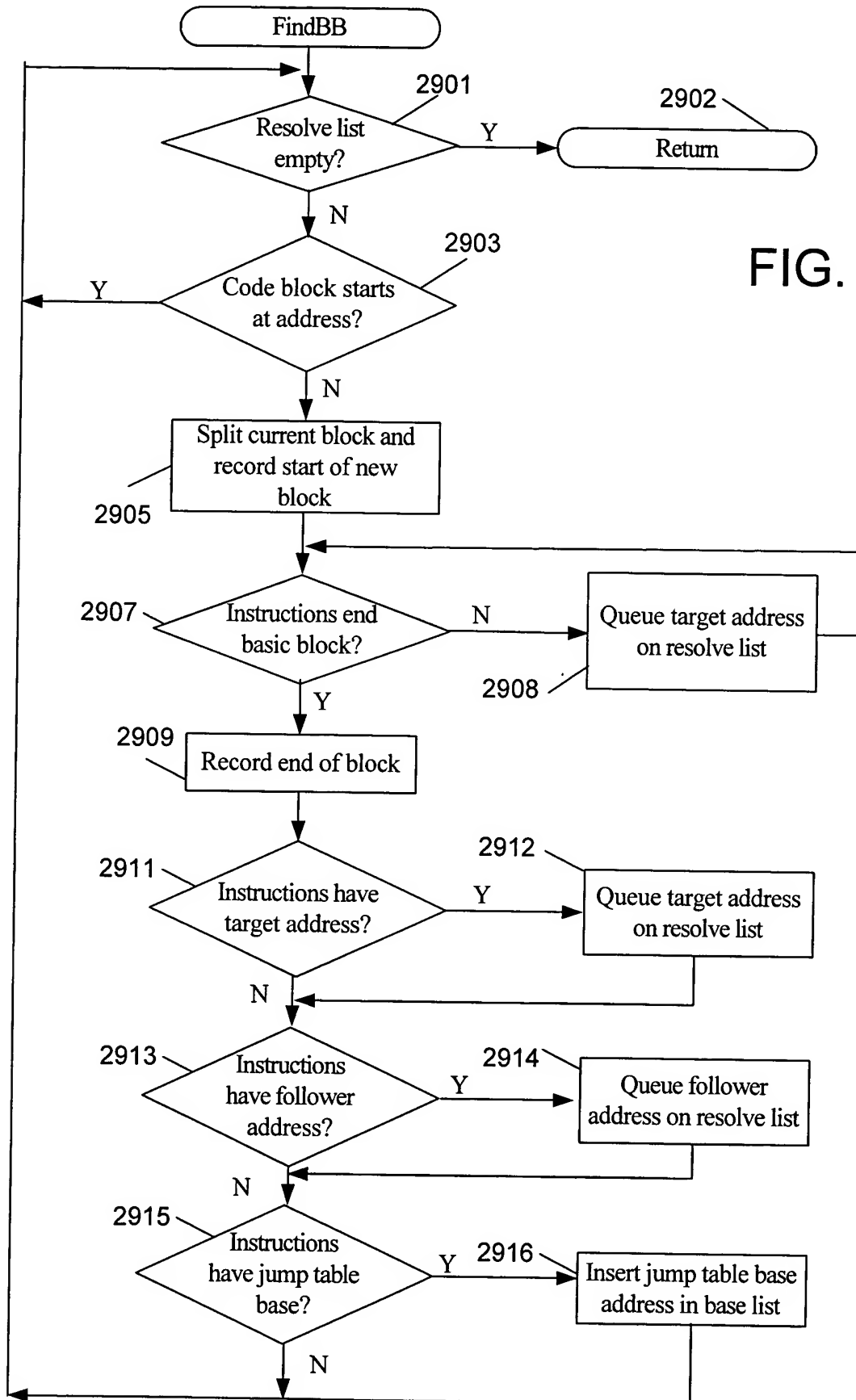


FIG. 29

FIG. 30

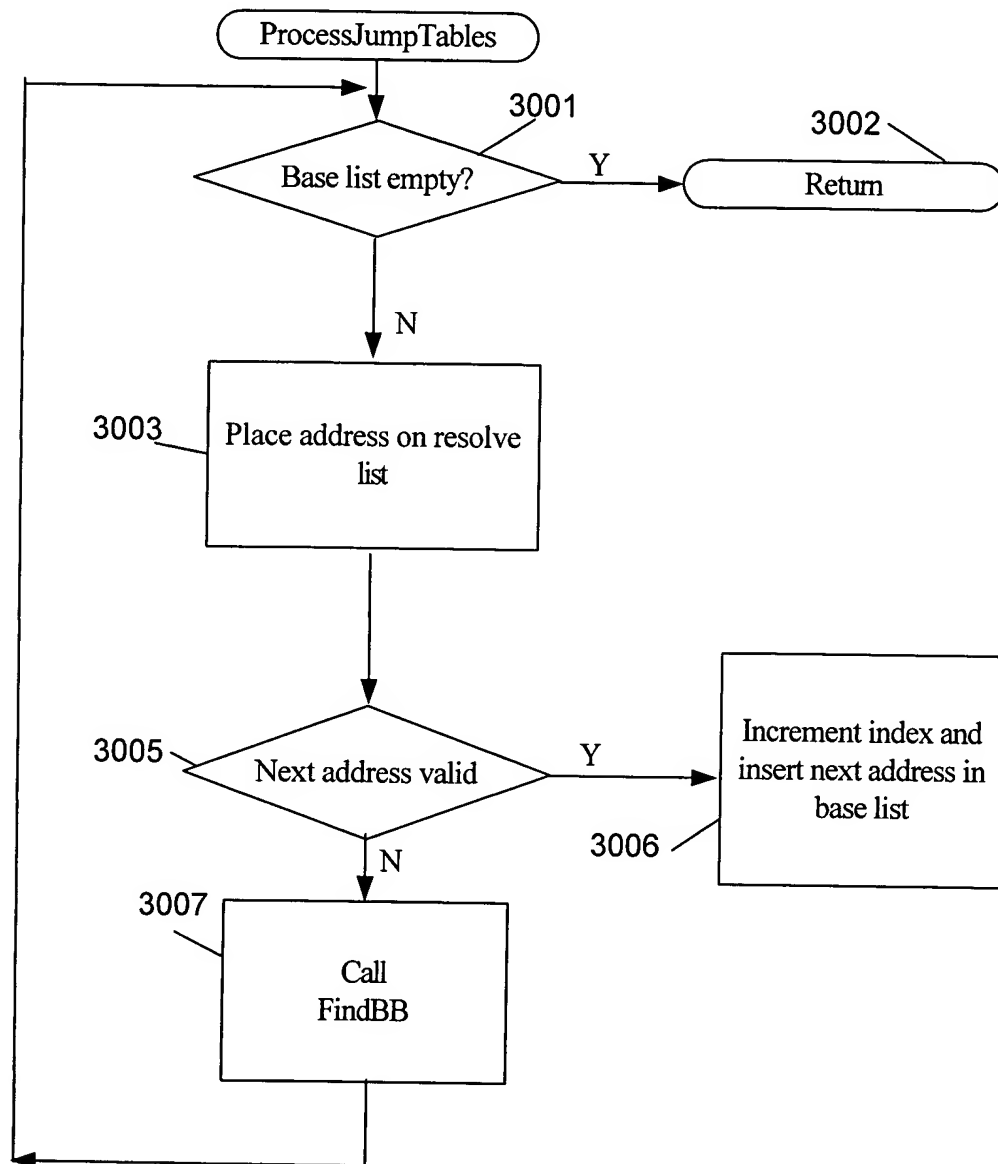


FIG. 31

